Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 384 (2007) 152–160
Theoretical Computer Science
www.elsevier.com/locate/tcs

A tight bound for online colouring of disk graphs

Ioannis Caragiannis[a,*], Aleksei V. Fishkin[b], Christos Kaklamanis[a], Evi Papaioannou[a]

[a] *Research Academic Computer Technology Institute and Department of Computer Engineering and Informatics, University of Patras, 26500 Rio, Greece*
[b] *Max Plank Institut für Informatik, Im Stadtwald, Geb. 46.1, 66123 Saarbrücken, Germany*

Abstract

We present an improved upper bound on the competitiveness of the online colouring algorithm First-Fit in disk graphs, which are graphs representing overlaps of disks on the plane. We also show that this bound is best possible for deterministic online colouring algorithms that do not use the disk representation of the input graph. We also present a related new lower bound for unit disk graphs.

Keywords: Graph colouring; Online algorithms; Disk graphs

1. Introduction

We study minimum colouring, a fundamental combinatorial optimization problem in graphs. Given a graph G, the minimum colouring problem is to find an assignment of colours (denoted by positive integers) to the nodes of the graph so that no two nodes connected by an edge are assigned the same colour and the number of colours used is minimized. We consider intersection graphs modelling overlaps of disks on the plane.

The intersection graph of a set of disks in the Euclidean plane is the graph having a node for each disk and an edge between two nodes if and only if the corresponding disks overlap. Each disk is defined by its radius and the coordinates of its center. Two disks overlap if the distance between their centres is at most equal to the sum of their radii. A graph G is called a *disk graph* if there exists a set of disks in the Euclidean plane whose intersection graph is G. The set of disks is called the disk representation of G. A disk graph is called *unit disk graph* if all disks in its disk representation have the same radius. A disk graph is σ*-bounded* if the ratio between the maximum and the minimum radius among all the disks in its disk representation is at most σ.

In disk graphs, minimum colouring is important since it can model frequency assignment problems in radio communication networks utilizing the Frequency Division Multiplexing technology [11]. Consider a set of transmitters located in fixed positions within a geographical region. Each transmitter may select to use a specific frequency from an available spectrum in order to transmit its messages. Two transmitters can successfully

* Corresponding author.
E-mail addresses: caragian@cti.gr (I. Caragiannis), avf@mpi-sb.mpg.de (A.V. Fishkin), kakl@cti.gr (C. Kaklamanis), papaioan@cti.gr (E. Papaioannou).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2007.04.025

(i.e. without signal interference) transmit messages simultaneously either if they use different frequencies or if they use the same frequency and their ranges do not overlap. Given a set of transmitters in a radio network, in order to guarantee successful transmissions simultaneously, the important engineering problem to be solved is the frequency assignment problem where the objective is to minimize the number of frequencies used all over the network. Assuming that all transmitters have circular range, the graph reflecting possible interference between pairs of transmitters is a disk graph. The frequency assignment problem is equivalent to minimum colouring.

An instance of the minimum colouring problem may or may not include the disk representation (i.e. disk centre coordinates and/or radii) of the disk graph as part of the input. Clearly, the latter case is more difficult. Information about the disk representation of a disk graph is not easy to extract. Actually, determining whether a graph is a disk graph is an NP-hard problem [12].

The minimum colouring problem has been proved to be NP-hard in [3,9] even for unit disk graphs. A naive algorithm is algorithm First-Fit: it examines the nodes of the graph in an arbitrary order and assigns to each node the smallest colour not assigned to its already examined neighbors. Clearly, First-Fit does not use the disk representation. It computes 5-approximate solutions in unit disk graphs [8,15]. By processing the nodes of the graph in a specific order, First-Fit computes 3-approximate solutions in unit disk graphs [9,15,16]. In general disk graphs, a smallest-degree-last version of First-Fit achieves an approximation ratio of 5 [8,14,15].

In the online versions of the problem, the disk graph is not given in advance but is revealed in steps. In each step, a node of the graph appears together with its edges incident to nodes that have appeared in previous steps (and possibly, together with the centre coordinates and/or the radius of the corresponding disk). When a node appears, an online colouring algorithm decides which colour to assign to the node. The decisions of the algorithm at a step cannot change in the future.

The performance of an online algorithm is measured in terms of its competitive ratio (or competitiveness, [1]) which is defined as the maximum over all possible sequences of disks of the ratio of the number of colours used by the algorithm over the minimum number of colours sufficient for coloring the graph (i.e. its chromatic number).

First-Fit is essentially an online algorithm. It has been widely studied in a more general context and has been proved to be $\Theta(\log n)$-competitive in inductive graphs with n nodes [13,10]. Disk graphs are inductive [5,6] so the upper bound holds for disk graphs as well. The lower bound holds also for trees (which are disk graphs) so the $\Theta(\log n)$ bound holds for general disk graphs. In unit disk graphs, First-Fit is known to be at least 4- [17] and at most 5-competitive [8,15] while for σ-bounded disk graphs with n nodes, it is known to be at most $O(\min\{\log n, \sigma^2\})$-competitive [5]. For unit disk graphs, a lower bound of 2 on the competitiveness of any deterministic online colouring algorithm is presented in [7]. The best known lower bound on the competitiveness of deterministic colouring algorithms in σ-bounded disk graphs with n nodes is $\Omega(\min\{\log n, \log\log\sigma\})$ [5]. Both lower bounds hold even for deterministic algorithms that use the disk representation. A competitive ratio of $O(\min\{\log n, \log\sigma\})$ is achieved by two algorithms presented in [5] and [2]. The former uses the disk representation while the latter does not but it is quite impractical. Both algorithms use First-Fit as a subroutine.

In this paper we show that algorithm First-Fit itself is $O(\log\sigma)$-competitive when applied to σ-bounded disk graphs. This significantly improves the previously known upper bound of $O(\sigma^2)$ on the competitiveness of First-Fit. Furthermore, it matches the best known upper bound for online deterministic colouring algorithms, previously achieved either by algorithms that use the disk representation [5] or by quite impractical algorithms that do not use the disk representation [2]. Our second result indicates that First-Fit has optimal competitiveness (within constant factors) among all deterministic online algorithms for disk graphs that do not use the disk representation. In particular, we show that any deterministic online colouring algorithm that does not use the disk representation has competitive ratio $\Omega(\log\sigma)$ on σ-bounded disk graphs. Combined with previous results of [5], our lower bound establishes a tight bound of $\Theta(\min\{\log n, \log\sigma\})$ on the optimal competitiveness of deterministic online coloring algorithms in σ-bounded disk graphs with n nodes that do not use the disk representation. We also prove a new lower bound of 2.5 on the competitiveness of deterministic online colouring algorithms for unit disk graphs that do not use the disk representation. This result improves a previous lower bound of 2 [7].

The rest of the paper is structured as follows: a discussion on previous upper bounds and the proof of the upper bound on the competitiveness of First-Fit are presented in Section 2. The lower bounds are presented in Section 3. We conclude with open problems in Section 4.

2. The upper bound

In this section, we prove the upper bound for algorithm First-Fit. Although this upper bound can also be achieved by two other known algorithms presented in [5] and [2], respectively, our result is important because of the simplicity of algorithm First-Fit.

The algorithm of Erlebach and Fiala [5] classifies the disks into a logarithmic number of classes so that the disks belonging to the same class form a 2-bounded disk graph and runs algorithm First-Fit in each class using disjoint sets of colours for colouring the disks of different classes. The classification is performed according to the radii of the disks; hence, the algorithm uses the disk representation. The proof of the $O(\log \sigma)$ upper bound follows by the fact that algorithm First-Fit has constant competitive ratio on 2-bounded disk graphs.

The algorithm Layered (which was presented in [2]) classifies the disks into layers and applies algorithm First-Fit to each layer separately, using a different set of colors in each layer. Layers are numbered with integers 1, 2, ... and a disk is classified into the smallest layer possible under the constraint that it cannot be classified into a layer if it overlaps with at least 16 mutually non-overlapping disks belonging to this layer.

The proof that algorithm Layered is $O(\log \sigma)$-competitive is based on the following arguments. First, if a disk of radius R belongs to some layer $i > 1$, then there is a disk of radius at most $R/2$ belonging to layer $i - 1$. Hence, if the disk graph given as input to algorithm Layered is σ-bounded, the number of layers is at most $1 + \log \sigma$. The logarithmic competitive ratio follows since the maximum independent set in the neighbourhood of each node within each layer has size at most 15, and, hence, algorithm First-Fit can be proved to have constant competitive ratio within each layer. Clearly, algorithm Layered does not use the disk representation.

Checking whether a new node presented has 16 or more non-overlapping disks of some layer in its neighbourhood may require time $\Omega(n^{16})$. This could be decreased to $\Omega(n^8)$ by changing the constraint so that a disk cannot be classified into a layer if it overlaps with at least 8 mutually non-overlapping disks belonging to this layer. Still, it can be proved that there exists a constant $\alpha > 1$ such that for each disk of radius R belonging to some layer $i > 1$, there exists a disk at layer $i - 1$ of radius smaller than R/α. This is the best possible improvement in the idea of algorithm Layered since, for any $\alpha > 1$ arbitrarily close to 1 (e.g., $\alpha = 1 + 1/\sigma$), a disk of radius R can overlap with 7 mutually non-overlapping disks of radius R/α, and, hence, the logarithmic upper bound on the number of layers cannot be established.

Surprisingly, we show that algorithm First-Fit itself is at most $O(\log \sigma)$-competitive, improving the previously known upper bound of $O(\sigma^2)$ [5]. Combining this result with the $O(\log n)$ upper bound which is known for the competitive ratio of First-Fit we obtain that First-Fit is $O(\min\{\log n, \log \sigma\})$-competitive. Algorithm First-Fit runs in time proportional to the number of edges of the disk graph (i.e. $O(n^2)$) and does not use the disk representation. Hence, it is much simpler than the previously known algorithms that achieve the same competitiveness bounds.

Theorem 1. First-Fit *is $O(\log \sigma)$-competitive for σ-bounded disk graphs.*

Proof. Let G be a σ-bounded disk graph with chromatic number κ. Assume that the nodes of G appear online and are colored by algorithm First-Fit. Consider a representation of G by overlapping disks on the plane of radii between r (the radius of the smallest disk) and $R_{\max}$ (the radius of the largest disk) so that $R_{\max}/r \leq \sigma$. We classify the nodes into levels $0, 1, \ldots, \lfloor \log (R_{\max}/r) \rfloor$ as follows: a node corresponding to a disk of radius R belongs to level $\lfloor \log (R/r) \rfloor$. Since $R_{\max}/r \leq \sigma$, the index of the last level is at most $\lfloor \log \sigma \rfloor$.

We will first show that a node of G belonging to level $i \geq 0$ is adjacent to at most $15(\kappa - 1)$ other nodes of level at least i.

Assume otherwise that there exists a node u of G at level i which is adjacent to at least $15\kappa - 14$ other nodes of level at least i. Let R be the radius of the disk d corresponding to node u in the disk representation. Then $i = \lfloor \log (R/r) \rfloor$. Also, let S_d be the set of disks corresponding to nodes adjacent to d which belong to levels at least i. Clearly, all the disks of S_d have radii at least $r2^i$.

We apply the following shrinking procedure on the disks of S_d. We shrink each disk d' in S_d into a disk of radius $r2^{\lfloor \log R/r \rfloor}$ as follows: If the center $c_{d'}$ of d' is inside d, we shrink d' into a disk having the same center $c_{d'}$. Otherwise, let $p_{d'}$ be the point in the periphery of d' which is closest to the center of d. We shrink d' so that $p_{d'}$ is again the point in the periphery of d' which is closest to the center of d. Denote by S'_d the set of shrunk disks. Clearly, each of the disks in S'_d overlaps with d since either its centre is contained in d or a point in its periphery is contained in d. This

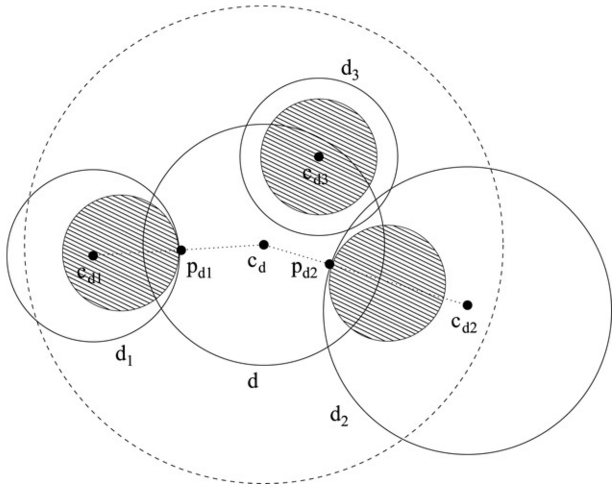

Fig. 1. The shrinking procedure. The disk d overlaps with three disks. Grey disks are the three corresponding shrunk disks.

means that all disks in S'_d are completely contained into the disk of radius $R + r2^{i+1}$ centered at the centre c_d of disk d. An example of the shrinking procedure is depicted in Fig. 1.

The node-induced subgraph H of G defined by the nodes of G corresponding to the disks of S_d is $(\kappa - 1)$-colourable since the graph G is κ-colourable and the nodes of H are all adjacent to u in G. Consequently, since by our assumption H contains the neighbours of u in G with levels at least i and since there are at least $15\kappa - 14$ such nodes, the maximum independent set in H has size at least 16. Consider such an independent set of 16 nodes in H and the 16 non-overlapping disks of S_d corresponding to these nodes. Clearly, the 16 corresponding shrunk disks of S'_d are also non-overlapping. Each of these disks has radius $r2^i$ and, hence, their total area is

$$\begin{aligned} 16\pi \left(r2^i\right)^2 &= \pi \left(r2^{i+1} + r2^{i+1}\right)^2 \\ &> \pi \left(R + r2^{i+1}\right)^2. \end{aligned}$$

Since these disks are non-overlapping, this contradicts the fact that all disks of S'_d are completely contained in the disk of radius $R + r2^{i+1}$ centered at c_d. Consequently, our assumption is incorrect and u is adjacent to at most $15(\kappa - 1)$ other nodes of G of level at least i.

We will now show that each node of G at level $i \geq 0$ is colored by algorithm First-Fit with a color in the range $[1, (15\kappa - 14)\,(i + 1)]$. Hence, the maximum color that can be assigned to a node of G by First-Fit is at most $(15\kappa - 14)\,(\lfloor \log \sigma \rfloor + 1)$. Since G has chromatic number κ, this implies that algorithm First-Fit is $O(\log \sigma)$-competitive.

We use induction on the level of nodes. The statement is true for nodes of level 0, since any such node is adjacent to at most $15(\kappa - 1)$ nodes of G and, hence, it will be assigned a colour in $[1, 15\kappa - 14]$.

Now assume that the statement is true for nodes of level $i = 0, \ldots, k$ (for $k < \lfloor \log \sigma \rfloor$). We will show that it also holds for nodes of level $k + 1$. Consider a node u at level $k + 1$. This node will be adjacent to nodes of smaller levels which may use colours up to color $(15\kappa - 14)(k + 1)$ and to at most $15(\kappa - 1)$ additional nodes of levels at least $k + 1$. Hence, the maximum colour that can be assigned by algorithm First-Fit to node u is $(15\kappa - 14)(k + 1) + 15(\kappa - 1) + 1 = (15\kappa - 14)(k + 2)$. This completes the proof of the theorem. $\square$

3. The lower bound

The result proved in the following establishes that algorithm First-Fit achieves the best possible competitive ratio (within constant factors) among all deterministic online coloring algorithms that do not use the disk representation.

We present an adversary $\mathcal{ADV}$ which, on input an integer k and a deterministic online coloring algorithm $\mathcal{A}$, outputs a forest $T_{\mathcal{A}}(k)$ with at most 2^{k-1} nodes such that $\mathcal{A}$ colours $T_{\mathcal{A}}(k)$ with at least k colours. We describe the adversary $\mathcal{ADV}$ in the following. This is a non-recursive description of an adversary used in [5] for proving lower bounds on

1. Create 2^{k-2} nodes labeled $\langle i, 1\rangle$ for each $i = 0, \ldots, 2^{k-2} - 1$, and introduce them to algorithm $\mathcal{A}$.
2. For $i = 2$ to $k - 1$ do
3.   For $j = 0$ to $2^{k-i-1} - 1$ do
4.   Set $S_\ell(j, i) = \{\langle j2^{i-1}, 1\rangle, \langle j2^{i-2}, 2\rangle, \ldots, \langle 2j, i - 1\rangle\}$
5.   Set $S_r(j, i) = \{\langle j2^{i-1} + 2^{i-2}, 1\rangle, \langle j2^{i-2} + 2^{i-3}, 2\rangle, \ldots, \langle 2j + 1, i - 1\rangle\}$
6.   Let $C_\ell(j, i)$ be the set of different colours assigned by $\mathcal{A}$ in the nodes of $S_\ell(j, i)$
7.   Let $C_r(j, i)$ be the set of different colors assigned by $\mathcal{A}$ in the nodes of $S_r(j, i)$
8.   If $C_\ell(j, i) = C_r(j, i)$ then
9.   Create a new node $\langle j, i\rangle$ connected to the nodes of $S_r(j, i)$ and introduce it to algorithm $\mathcal{A}$.
10.   else
11.   Let $\langle s, t\rangle$ be a node of $S_r(j, i)$ to which $\mathcal{A}$ assigns a color not in $C_\ell(j, i)$.
12.   Rename $\langle s, t\rangle$ as $\langle j, i\rangle$
13. Create a new node r connected to nodes of $S_\ell(0, k - 1) \cup \{\langle 0, k - 1\rangle\}$ and introduce it to algorithm $\mathcal{A}$.

Fig. 2. The adversary $\mathcal{ADV}$.

disk graphs and (in a more general form) in [13] for proving lower bounds on inductive graphs. The adversary assigns levels from 1 to k to the nodes. For representing the nodes in the forest except the node of level k, we use the notation $\langle j, i\rangle$, where i is an integer representing the level of the node ($1 \le i \le k - 1$) and j is an integer that identifies the node among the nodes of the same level ($0 \le j \le 2^{k-i-1} - 1$).

The description of the adversary $\mathcal{ADV}$ is depicted in Fig. 2. The adversary forces the algorithm to use at least k colors. In each iteration, we will show by induction on i that all the $i - 1$ nodes examined for defining the set $S_\ell(j, i)$ (and similarly for $S_r(j, i)$) are coloured with $i - 1$ different colours. Hence, after the if-then-else statement, the adversary will have forced algorithm $\mathcal{A}$ to use i different colours, i.e. $k - 1$ colours at the end of all iterations. This clearly holds if the sets $S_\ell(j, i)$ and $S_r(j, i)$ are not the same (else statement in line 10). Otherwise, it is guaranteed by the introduction of a new node (line 9) which is connected to nodes coloured with the $i - 1$ different colors of $S_r(j, i)$ (if-then statement). Then, the last node r is connected to nodes with $k - 1$ different colours (line 13) and will be assigned a k-th colour by algorithm $\mathcal{A}$. Formally, we prove the following claim:

Claim 2. *On input an integer k and a deterministic online colouring algorithm $\mathcal{A}$, the adversary $\mathcal{ADV}$ produces a graph $T_{\mathcal{A}}(k)$ which $\mathcal{A}$ colours with at least k colours.*

Proof. We will first show inductively that for any $i = 2, \ldots, k - 1$, the number of colours used in the nodes of the sets $S_\ell(j, i)$ and $S_r(j, i)$ for any $j = 0, \ldots, 2^{k-i-1} - 1$ is $i - 1$. This is clearly true for $i = 2$ and for any $j = 0, \ldots, 2^{k-3} - 1$ since in this case the sets $S_\ell(j, i)$ and $S_r(j, i)$ are singletons.

Assume that the statements are true for $i = 2, \ldots, i'$ for $2 \le i' < k - 1$ (and any j such that $0 \le j \le 2^{k-i-1} - 1$). We will show that they are also true for $i = i' + 1$. Let j be such that $0 \le j \le 2^{k-i'-2} - 1$. We first show the case for $S_\ell(j, i' + 1)$. Observe that $S_\ell(j, i' + 1) = S_\ell(2j, i') \cup \{\langle 2j, i'\rangle\}$. By the inductive hypothesis, the nodes of the sets $S_\ell(2j, i')$ and $S_r(2j, i')$ have been assigned $i' - 1$ different colours by algorithm $\mathcal{A}$. Consider the iteration of the adversary with values i' and $2j$. By the definition of the adversary, if the set $C_\ell(2j, i')$ of $i' - 1$ colors assigned to the nodes of $S_\ell(2j, i')$ is the same as the set $C_r(2j, i')$ of $i' - 1$ colours assigned to the nodes of $S_r(2j, i')$ (if-then statement in lines 8–9), the adversary introduces a new node $\langle 2j, i'\rangle$ to algorithm $\mathcal{A}$ which is connected to all nodes of $S_r(2j, i')$. Hence, the algorithm $\mathcal{A}$ will use a colour different from the $i' - 1$ colours of $C_\ell(2j, i')$ to colour the new node $\langle 2j, i'\rangle$, and i' different colours will have been assigned to the nodes of $S_\ell(j, i' + 1)$. Otherwise, if the $i' - 1$ colours in each of the sets $C_\ell(2j, i')$ and $C_r(2j, i')$ are not the same (else statement in lines 10–12), the adversary renames a node of $S_r(2j, i')$ that has been assigned a colour different from the $i' - 1$ colors assigned to the nodes in $S_\ell(2j, i')$ as $\langle 2j, i'\rangle$. Hence, in this case, i' different colours will have been assigned to the nodes of $S_\ell(j, i' + 1)$.

For proving that the statement is true for $S_r(j, i'+1)$, we follow the same arguments, the only difference being that $S_r(j, i'+1) = S_\ell(2j+1, i') \cup \{\langle 2j+1, i'\rangle\}$.

At the last iteration of the adversary, the node labelled $\langle 0, k-1\rangle$ is either introduced or renamed. Using the same reasoning as above, the algorithm colours this node with a color different from the $k-2$ colours used in the nodes of $S_\ell(0, k-1)$. Hence, when the last node is introduced (line 13), it has to be coloured with a k-th colour since it is connected to all nodes of the set $S_\ell(0, k-1) \cup \langle 0, k-1\rangle$. □

Next, we show that when a new node is introduced in an iteration, the nodes to which it is connected will not be connected to other nodes in subsequent iterations. Hence, in general, the resulting graph is a forest. Since there are 2^{k-2} nodes at level 1, at most one new node is introduced in each iteration (line 9), and one more node is introduced at the end (line 13), the number of nodes is at most

$$2^{k-2} + \sum_{i=2}^{k-1} 2^{k-i-1} + 1 = 2^{k-1}.$$

Claim 3. *For any integer k and any deterministic online colouring algorithm $\mathcal{A}$, the graph $T_{\mathcal{A}}(k)$ produced by the adversary $\mathcal{ADV}$ is a forest.*

Proof. We will show that a node can be connected to at most one node of higher level. Let $\langle j, i\rangle$ be the label of a node the last time it is renamed by the adversary (or the first label assigned to it if it is never renamed). Note that when a node is renamed it is not connected to any node of higher levels. Assume that $\langle j, i\rangle$ belongs to the set $S_r(j', i')$ for some i' and j' such that $i < i' \leq k-1$ and $0 \leq j' \leq 2^{i'-2} - 1$. By the definition of the set $S_r(j', i')$, we have that $j = 2^{i'-i} j' + 2^{i'-i-1}$. This is the only possibility for this node to be connected to higher levels and in the case that it is connected, it keeps its label at the next step. So, assume that at some later execution, node $\langle j, i\rangle$ belongs to the set $S_r(j'', i'')$ for $i'' \geq i'$. Again, it must be $j = 2^{i''-i} j'' + 2^{i''-i-1}$, so we obtain that $2^{i''-i} j'' + 2^{i''-i-1} = 2^{i'-i} j' + 2^{i'-i-1} \Rightarrow 2^{i''-i'+1} j'' + 2^{i''-i'} = 2j' + 1$. Observe that this last equality can only hold if $i'' = i'$ and in this case it must also be $j'' = j'$, otherwise the left part of the equality is even and the right is odd. So, after it is connected to some node of higher level, node $\langle j, i\rangle$ will never belong to another set $S_r(j'', i'')$, so it will never be renamed again and will never be connected to some other node of higher level. □

When the adversary runs against algorithm First-Fit, then the constructed graph $T_{FF}(k)$ has exactly 2^{k-1} nodes. This is because, by its definition and by the definition of the adversary, algorithm First-Fit colours the nodes of level i with colour i. So, in each iteration, the sets $C_\ell(j, i)$ and $C_r(j, i)$ have the same $i-1$ colours, so the adversary never executes the else statement at lines 10–12 and a new node is introduced (lines 8–9) which the algorithm colours with colour i. We formally show that $T_{FF}(k)$ is actually a tree since each node is connected to some node of higher level. In fact, $T_{FF}(k)$ is a *binomial tree*; such trees have been used in many contexts in the literature to prove logarithmic bounds (e.g. [4], p. 457).

Claim 4. *For any $i = 1, \ldots, k-1$ and j such that $0 \leq j \leq 2^{k-i-1} - 1$, node $\langle j, i\rangle$ is connected to a node of higher level.*

Proof. First observe that at the last step the algorithm connects all nodes $\langle 0, i\rangle$ for $i = 1, \ldots, k-1$ to the last node. So, it suffices to show that each node $\langle j, i\rangle$ with $1 \leq i \leq k-2$ and $1 \leq j \leq 2^{k-i-1} - 1$ belongs to the set $S_r(j', i')$ for some i' and j' such that $i < i' \leq k-1$ and $0 \leq j' \leq 2^{k-i'-1} - 1$. Let z be the largest non-negative integer such that j is divisible by 2^z. Since $1 \leq j \leq 2^{k-i-1} - 1$, it is $0 \leq z \leq k-i-2$. Set $i' = i+z+1$. Then, $i+1 \leq i' \leq k-1$. Also, $j - 2^z$ is divisible by 2^{z+1} and $\frac{j-2^z}{2^{z+1}} \leq \frac{2^{k-i-1}}{2^{z+1}} - 1 = 2^{k-(i+z+1)-1} - 1 = 2^{k-i'-1} - 1$. Set $j' = \frac{j-2^z}{2^{z+1}}$. Since $j = j' 2^{z+1} + 2^z = j' 2^{i'-i} + 2^{i'-i-1}$, node $\langle j, i\rangle$ belongs to set $S_r(j', i')$ and will be connected to node $\langle j', i'\rangle$. □

An example of the tree $T_{FF}(k)$ is depicted in Fig. 3. In the following, we will first show that $T_{FF}(k)$ is an α^{k-1}-bounded disk graph, for every $\alpha > 2$. Then, we will show how to adapt this construction for forests produced by the adversary against other deterministic online colouring algorithms.

Given a disk d of radius R corresponding to some node of the tree $T_{FF}(k)$, we define the vertical strip of d to be the vertical strip of width $2R$ which completely contains d. In our construction, the disk representation of $T_{FF}(k)$ is such that the disks corresponding to nodes in the subtree of a node u do not cross the boundaries of the vertical strip

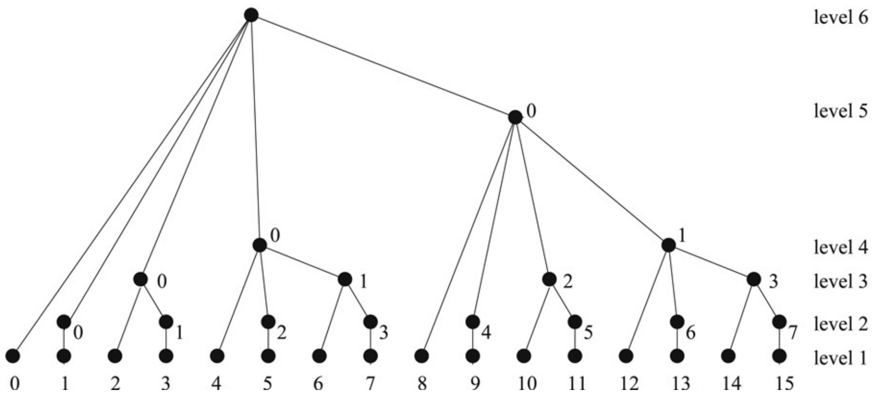

Fig. 3. The tree $T_{FF}(6)$.

of the disk d corresponding to u. Furthermore, the vertical strips of any two disks corresponding to children of the same node are disjointed. These two invariants guarantee that the disks corresponding to nodes belonging to different subtrees do not overlap.

We first locate a disk of radius α^{k-1} corresponding to the root r of the tree. Disks corresponding to nodes of level i (for $i = 1, \ldots, k-1$) will have radius α^{i-1}.

A node u at level i with $i = 2, \ldots, k$, has $i-1$ children $u_1, \ldots, u_{i-1}$ in $T_{FF}(k)$ with levels $1, \ldots, i-1$, respectively. Let d be the disk corresponding to node u and let $d_1, \ldots, d_{i-1}$ be the disks corresponding to its children $u_1, \ldots, u_{i-1}$, respectively. Assuming that the centre of the disk d has horizontal coordinate h, the centre of the disk d_j has horizontal coordinate $h_j = h - \alpha^{i-1} + 3\alpha^{i-1}2^{-i+j}$. The vertical strips of the disks $d_1, \ldots, d_{i-1}$ are disjointed since for two disks d_j and $d_{j'}$ with $j > j'$, their centres differ in the horizontal coordinate by

$$\begin{aligned} h_j - h_{j'} &= (h - \alpha^{i-1} + 3\alpha^{i-1}2^{-i+j}) - (h - \alpha^{i-1} + 3\alpha^{i-1}2^{-i+j'}) \\ &= \frac{3}{2}\alpha^{j-1}(\alpha/2)^{i-j}(2 - 2^{-j+j'+1}) \\ &> \frac{3}{2}\alpha^{j-1} \\ &> \alpha^{j-1} + \alpha^{j'-1} \end{aligned}$$

which is the sum of their radii.

Furthermore, neither the leftmost disk d_1 nor the rightmost disk d_{i-1} cross the boundary of the vertical strip of d. Indeed, the leftmost point of d_1 has horizontal coordinate

$$\begin{aligned} h_1 - 1 &= h - \alpha^{i-1} + 3\alpha^{i-1}2^{-i+1} - 1 \\ &= h - \alpha^{i-1} + 3(\alpha/2)^{i-1} - 1 \\ &> h - \alpha^{i-1} + 2 \\ &> h - \alpha^{i-1} \end{aligned}$$

which is the horizontal coordinate of the left boundary of the vertical strip of d. Also, the rightmost point of d_{i-1} has horizontal coordinate

$$\begin{aligned} h_{i-1} + \alpha^{i-2} &= h - \alpha^{i-1} + 3\alpha^{i-1}/2 + \alpha^{i-2} \\ &= h + \alpha^{i-1}/2 + \alpha^{i-2} \\ &< h + \alpha^{i-1} \end{aligned}$$

which is the horizontal coordinate of the right boundary of the vertical strip of d.

The vertical coordinate of the centre of disk d_j is defined so that it is smaller than the vertical coordinate of the lowest point in the intersection of disk d_j with disk d. This guarantees that, among all disks corresponding to nodes in the subtree of u, the disks that d overlaps with are those corresponding to its children in $T_{FF}(k)$.

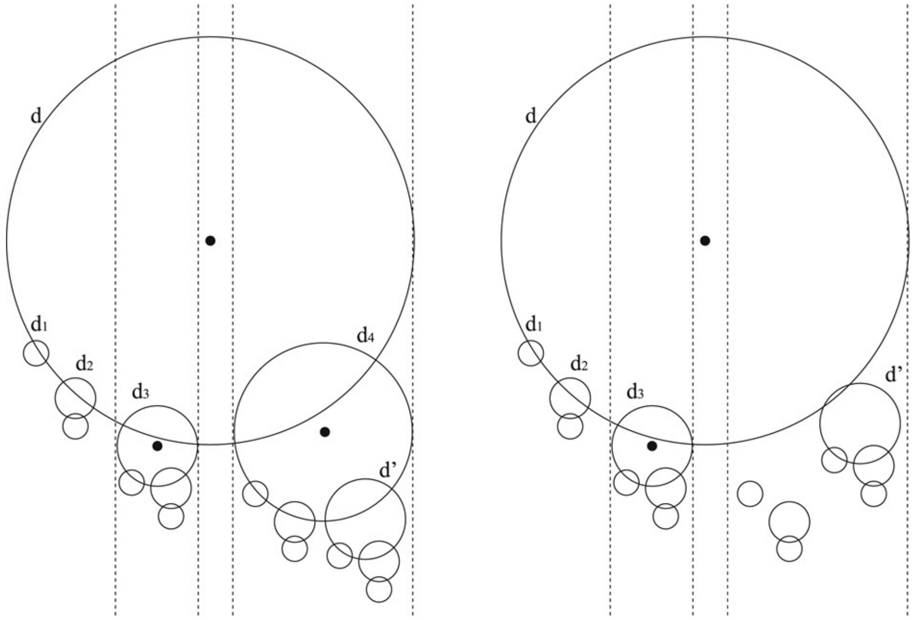

Fig. 4. (a) The disk representation of the tree $T_{FF}(5)$. The dashed lines indicate the boundaries of the vertical strips of two disks. (b) The disk representation of the forest produced by the adversary $\mathcal{ADV}$ when the node corresponding to disk d_4 is not introduced and the node corresponding to disk d' is renamed.

In the disk representation of the tree $T_{FF}(k)$, we use disks of radii between 1 and α^{k-1}. So, $T_{FF}(k)$ is an α^{k-1}-bounded disk graph. An example of the construction is depicted in the left part of Fig. 4.

Now, consider the forest created by the adversary on input an integer k and some other algorithm $\mathcal{A}$. In this case, some iterations may have renamed some nodes instead of introducing new ones. We construct the disk representation of such a forest by starting with the disk representation of $T_{FF}(k)$. We follow the execution of the adversary on input k and some algorithm $\mathcal{A}$. When, the adversary executes the else statement in an iteration associated with i and j, we remove the disk d_i corresponding to the node $\langle j, i\rangle$ in $T_{FF}(k)$ (since this node is not introduced) and move vertically all the disks corresponding to nodes of the subtree of node $\langle s, t\rangle$ so that the disk d' corresponding to node $\langle s, t\rangle$ (and no other disk in its subtree) overlaps with the disk corresponding to the parent node of node $\langle j, i\rangle$ in $T_{FF}(k)$. From now on, the node to which disk d' corresponds has been renamed as $\langle j, i\rangle$. An example is depicted in the right part of Fig. 4.

Clearly, this also yields an α^{k-1}-bounded disk graph. The above discussion leads to the following lemma:

Lemma 5. *For any $\alpha > 2$, the forest constructed by the adversary $\mathcal{ADV}$ on input an integer $k \geq 3$ and any deterministic online colouring algorithm $\mathcal{A}$ is an α^{k-1}-bounded disk graph.*

Now given a sufficiently large σ, the graph produced by the adversary on input $k = 1 + \lfloor \log_\alpha \sigma \rfloor$ and any deterministic algorithm is a 2-colourable σ-bounded disk graph which the algorithm colours with at least $1 + \lfloor \log_\alpha \sigma \rfloor$ colours. We obtain the following:

Theorem 6. *Any deterministic online algorithm for colouring σ-bounded disk graphs that does not use the disk representation has competitive ratio $\Omega(\log \sigma)$.*

For unit disk graphs, the best known lower bound on the competitiveness of any deterministic algorithm is 2 [7] and holds also for algorithms that use the disk representation. On input a deterministic online algorithm $\mathcal{A}$, the adversary in the proof of [7] constructs a κ-colorable unit disk graph with $\kappa \in \{1, 2, 3\}$, which algorithm $\mathcal{A}$ colours with at least 2κ colours. In the following, we improve this lower bound for deterministic online colouring algorithm in unit disk graphs that do not use the disk representation.

Consider a deterministic online colouring algorithm $\mathcal{A}$ and the forest produced by adversary $\mathcal{ADV}$ on input 5 and algorithm $\mathcal{A}$. Each connected component of the forest produced by $\mathcal{ADV}$ is a subtree of the tree $T_{FF}(5)$ produced by $\mathcal{ADV}$ on input 5 and algorithm First-Fit. The tree $T_{FF}(5)$ is a unit disk graph as shown in Fig. 5.

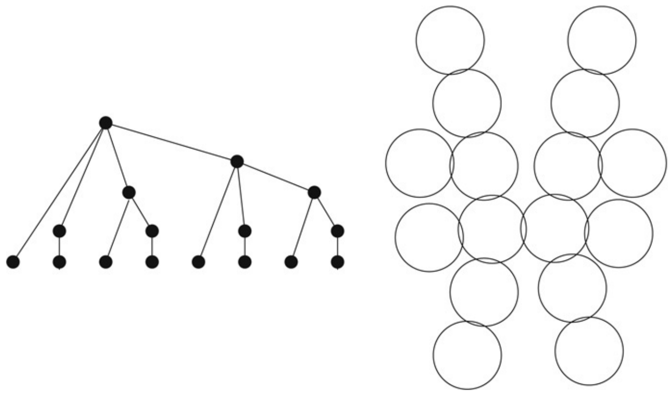

Fig. 5. The tree $T_{FF}(5)$ and its disk representation with unit disks.

Hence, the output of the adversary $\mathcal{ADV}$ on input 5 and any deterministic algorithm $\mathcal{A}$ is a unit disk graph. Since this output is a forest, it is 2-colourable, while the adversary forces $\mathcal{A}$ to use at least 5 colours. We obtain the following:

Theorem 7. *Any deterministic online algorithm for colouring unit disk graphs that does not use the disk representation has competitive ratio at least* 2.5.

4. Open problems

Our results on σ-bounded disk graphs can be extended to other classes of geometric graphs such as intersection graphs of squares and intersection graphs of rectangles whose height to width ratio is bounded by a constant. It still remains to show whether there exist deterministic online colouring algorithms that use the disk representation and have competitive ratio $o(\log \sigma)$. Our lower bound does not apply to this case.

Also, it would be interesting to investigate whether randomization helps in improving the known upper bounds and even beating the lower bounds for deterministic algorithms. To our knowledge, randomized online colouring algorithms have not been studied except for very general classes of graphs (e.g. in [18]) where the results are inherently much weaker than those for disk graphs.

References

[1] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.
[2] I. Caragiannis, A.V. Fishkin, C. Kaklamanis, E. Papaioannou, Online algorithms for disk graphs, in: Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science, MFCS '04, in: LNCS, vol. 3153, Springer, 2004, pp. 215–226.
[3] B.N. Clark, C.J. Colbourn, D.S. Johnson, Unit disk graphs, Discrete Mathematics 86 (1990) 165–177.
[4] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd ed., McGraw-Hill & MIT Press, 2001.
[5] T. Erlebach, J. Fiala, On-line coloring of geometric intersection graphs, Computational Geometry: Theory and Applications 9 (1–2) (2002) 3–24.
[6] T. Erlebach, J. Fiala, Independence and coloring problems on intersection graphs of disks, in: E. Bampis, et al. (Eds.), Efficient Approximation and Online Algorithms, in: LNCS, vol. 3484, Springer, 2006, pp. 135–155.
[7] J. Fiala, A.V. Fishkin, F.V. Fomin, On distance constrained labeling of disk graphs, Theoretical Computer Science 326 (1–3) (2004) 261–292.
[8] A. Gräf, Coloring and recognizing special graph classes, Musikinformatik und Medientechnik Bericht 20/95, Johannes Gutenberg-Universität Mainz, 1995.
[9] A. Gräf, M. Stumpf, G. Weissenfels, On coloring unit disk graphs, Algorithmica 20 (3) (1998) 277–293.
[10] A. Gyárfás, J. Lehel, On-line and first fit colorings of graphs, Journal of Graph Theory 12 (2) (1988) 217–227.
[11] D.K. Hale, Frequency assignment: Theory and applications, Proceedings of the IEEE 68 (12) (1980) 1497–1514.
[12] P. Hliněný, J. Kratochvíl, Representing graphs by disks and balls, Discrete Mathematics 229 (1–3) (2001) 101–124.
[13] S. Irani, Coloring inductive graphs on-line, Algorithmica 11 (1994) 53–72.
[14] E. Malesińska, Graph theoretical models for frequency assignment problems, Ph.D. Thesis, Technical University of Berlin, 1997.
[15] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi, D.J. Rosenkrantz, Simple heuristics for unit disk graphs, Networks 25 (1995) 59–68.
[16] R. Peeters, On coloring j-unit sphere graphs, Technical Report, Dept. of Economics, Tilburg University, 1991.
[17] Y.-T. Tsai, Y.-L. Lin, F.R. Hsu, The on-line first-fit algorithm for radio frequency assignment problems, Information Processing Letters 84 (2002) 195–199.
[18] S. Vishwanathan, Randomized online coloring of graphs, in: Proc. of the 31st Annual Symposium on Foundations of Computer Science, FOCS '90, 1990, pp. 464–469.